„Let me subscribe“
Zabbix masters IoT Topics

IntelliTrend IT-Services GmbH
Otto-Brenner-Strasse 119
D-33607 Bielefeld, Germany

Contact: Wolfgang Alper
wolfgang.alper@intellitrend.de
www.intellitrend.de
„Let me subscribe“
What is this all about

Provide MQTT as item type

https://support.zabbix.com/browse/ZBXNEXT-3950

For „Modbus“ support, see also:
https://support.zabbix.com/browse/ZBXNEXT-6090
https://support.zabbix.com/browse/ZBXNEXT-6093
„Let me subscribe“
What is this all about

Cool, but what is MQTT?
„Let me subscribe“
MQTT basics

- MQTT = „Message Queuing Telemetry Transport“.  
- Invented in 1999, designed to be bandwidth-efficient and lightweight, thus battery efficient.  
- Rapidly increasing adoption because of its suitability for Internet of Things (IoT), sensor networks, home automation, machine-to-machine (M2M) and mobile applications.  
- Usually runs over TCP/IP, Ports 1883 / 8883 (encrypted).  
- Encryption requires TLS transport.
„Let me subscribe“
MQTT basics

- Variation MQTT-SN (MQTT for Sensor Networks):
  Variant for non-TCP/IP networks like Zigbee (IEEE 80215.4 radio based protocol) or other UDP / Bluetooth based implementations.
- Based on a „publish“ / „subscribe“ to „topic“ mechanism.
- 2 types of network entities: „Message broker“ and „Clients“.
- Supports 3 QoS levels:
  - 0: At most once – „Fire and forget“.
  - 1: At least once – Could be sent/delivered multiple times.
  - 2: Exactly once – Safest and slowest service.
“Let me subscribe“
Publish / subscribe

Client-1
Publish
office/temp:25

Client-2
Publish
office/hum:42
office/iaq:110

MQTT Message Broker

Subscribe
office/temp
office/hum

Client-3
Publish
office/temp:25
office/hum:42

Subscribe
office/hum
office/iaq

Client-4
Publish
office/hum:42
office/iaq:110

Topic 1 (2 levels): office/temp
Topic 2 (2 levels): office/hum
„Let me subscribe“
Combined pub/sub

Client-1
- Publish
  - office/brightness:1500

Client-2
- Subscribe
  - office/lightctr
- Publish
  - office/lightctr:incr

MQTT Message Broker

Client-3
- Subscribe
  - office/brightness
- Publish
  - office/brightness:1500
- Publish
  - office/lightctr:incr
„Let me subscribe“
Wildcards subs

Client-1
- Publish
  office/confroom/brightness:1500

Client-2
- Publish
  office/kitchen/brightness:1600
  office/kitchen/temp:24

Client-3
- Subscribe
  office/+/
  brightness/
- Publish
  office/confroom/brightness:1500
  office/kitchen/brightness:1600

Client-4
- Subscribe
  office/kitchen/#
- Publish
  office/kitchen/brightness:1600
  office/kitchen/temp:24

MQTT Message Broker

+ = single level
# = multi level
„Let me subscribe“
MQTT - Summary

- Clients can publish and subscribe to one or more topics.
- One client can publish and subscribe at the same time.
- Clients can subscribe using single/multi level wildcards.
- Clients can choose between three different QoS levels.

Advanced features:
- Messages can be retained on the broker for new subscribers.
- Clients can provide a “last will and testament” that will be published by the broker when the client “dies”.

© Copyright 2020 IntelliTrend GmbH ● Germany ● www.intellitrend.de
„Let me subscribe“
What is this all about

Ok, but where does Zabbix come into play?
„Let me subscribe“
Real world example – Simple Setup

Server Room
Training Room

Published Topics

office/bielefeld/serverroom
office/bielefeld/trainingroom

office/bielefeld/salesroom
office/bielefeld/supportroom

Publish

mosquitto: https://mosquitto.org/

Subscribe

Home Automation
„Let me subscribe“
Real world example – Simple Setup

Server Room

Training Room

Published Topics

office/bielefeld/serverroom
office/bielefeld/trainingroom

Sales Room

Support Room

Published Topics

office/bielefeld/salesroom
office/bielefeld/supportroom

Zabbix 5.2 Agent 2

Subscribe

Publish

MQTT Message Broker

Zabbix as a subscriber
“Let me subscribe”
Real world example – Simple Setup

Server Room  Training Room

office/bielefeld/serverroom
office/bielefeld/trainingroom

Published Topics

office/bielefeld/salesroom
office/bielefeld/supportroom

Multiple clients

Zabbix 5.2 Agent 2

Zabbix

MQTT
Message Broker

Mosquitto
“Let me subscribe“
Real world example – Flow handling

Published Topics
- office/bielefeld/serverroom
- office/bielefeld/trainingroom

Published Topics
- office/bielefeld/salesroom
- office/bielefeld/supportroom

MQTT
Message Broker

Node-RED
Flow-based programming tool

Data Processing in Node-RED
“Let me subscribe”
Real world example – Flow handling

Server Room
Training Room

Published Topics
office/bielefeld/serverroom
office/bielefeld/trainingroom

office/bielefeld/salesroom
office/bielefeld/supportroom

Zabbix publishes data to broker

MQTT
Message Broker

Node-RED
Flow-based programming tool

Zabbix 5.2 Agent 2

Home Automation
„Let me subscribe“
Real world example – Flow handling

Published Topics
office/bielefeld/serverroom
office/bielefeld/trainingroom
office/bielefeld/salesroom
office/bielefeld/supportroom

Multiple Zabbix servers share same data
Let me subscribe“
What is Node-RED?

- Construction kit for the Internet of Things and home automation.
- Acts as MQTT client, can publish and subscribe.
- Flow-based tool for visual programming based on Node.js.
- Graphical web editor.
- Supports input, processing and output nodes.
- Extensible with plugins and custom function nodes.
„Let me subscribe“
Node-RED – Simple Subscription

Node-RED: https://nodered.org/
„Let me subscribe“
Sensor data

Raw JSON string

```
{ "hasenv":1, "haslaq":1, "hastempe":0, "temp":20.71, "hum":47.12793, "dew":9.490333, "press":99869.12, "eco2":136, "rsrs":77, "voc":4.235063, "batterycharge":100, "serial":"040EBBEF7F13C", "version":"1.5.0", "enverror":0, "laqerror":0, "lagerrorstatus":0, "laqheatingtime":1200, "laqbaseline":38007, "laq_enabled":0, "temerror":0, "sfails":0, "wfail":0, "cfails":0, "tfail":0, "wtime":3.174, "ctime":0, "reason":5, "warmreason":6, "interval":10, "location":"Sales Room", "contact":"IntelliTrend Monitoring Team", "geo_local":"52.013798", "geo_long":"8.56542", "device_type":"IM-Smart", "otaconfigstatus":0, "otaconfiginterval":24, "otafwstatus":0, "otafwinterval":168}
```

Parsed JSON object

```
{ "hasenv":1, "haslaq":1, "hastempe":0, "temp":20.71, "hum":47.12793, "dew":9.490333, "press":99869.12, "eco2":136, "rsrs":77, "voc":4.235063, "batterycharge":100, "serial":"040EBBEF7F13C", "version":"1.5.0", "enverror":0, "laqerror":0, "lagerrorstatus":0, "laqheatingtime":1200, "laqbaseline":38007, "laq_enabled":0, "temerror":0, "sfails":0, "wfail":0, "cfails":0, "tfail":0, "wtime":3.174, "ctime":0, "reason":5, "warmreason":6, "interval":10, "location":"Sales Room", "contact":"IntelliTrend Monitoring Team", "geo_local":"52.013798", "geo_long":"8.56542", "device_type":"IM-Smart", "otaconfigstatus":0, "otaconfiginterval":24, "otafwstatus":0, "otafwinterval":168}
```

Multiple metrics in one message
„Let me subscribe“

Options to process multiple metrics

- **Split on MQTT level:**
  Use Node-RED to split metrics and then publish them in their own topics. Good if there are other clients that can handle only a single metric at a time.

- **Split on Zabbix level:**
  Use Zabbix JSON pre-processing with corresponding dependant items. Zabbix would just need one subscription.

- **Combine both methods:**
  Let other clients subscribe to a single metric using their specific topic, but also publish all sensor data for Zabbix in one topic.
“Let me subscribe”  
Split metrics on MQTT level

Node setup

- Using JSONata expressions to extract metrics.
- Append sensor name to original topic.
- Publish to new topic in node message.

Node properties (temperature)

Debug output

Splitting data in Node-RED
„Let me subscribe“
Implementation

Zabbix side of things ...
„Let me subscribe“
How to use the new MQTT key

```python
mqtt.get[<broker_url>, topic, <username>, <password>]
```

- Requires Agent 2.
- Requires active checks.
- Broker URL default is localhost.
- User name and password are optional.
- Uses Eclipse Paho Go client library.

ZABBIX FEATURE REQUESTS / ZBXNEXT-3950
Provide MQTT as item type
„Let me subscribe“
Agent active - send to multiple hosts

One Zabbix agent active needs to send data to multiple hosts ...

Mapping: Topics to Zabbix hosts

Zabbix 5.2 Agent 2

office/bielefeld/salesroom
office/bielefeld/serverroom
office/bielefeld/supportroom
office/bielefeld/trainingroom

im.ims-040B0BBF713C
im.ims-603E9BC40A24
im.ims-80D478BF713C
im.ims-94B71112CFA4

Hostname=im.ims-040B0BBF713C,im.ims-603E9BC40A24,im.ims-80D478BF713C,im.ims-94B71112CFA4

Zabbix 5.2 Server

Running active agent checks from multiple Zabbix hosts

Zabbix FEATURE REQUESTS / ZBXNEXT-2943
“Let me subscribe“
Split metrics on Zabbix level

Master Item
- **Name**: Sensor data master
- **Type**: Zabbix agent (active)
- **Key**: mqtt.get(SMOTT_BROKER),(SMOTT_TOPIC),(SMOTT_USER),(SMOTT_PASS)
- **Update interval**: 1m

Dependent Item
- **Name**: Temperature (°C)
- **Type**: Dependent item
- **Key**: temp
- **Master item**: MS-Smart Sensor MQTT: Sensor data master
- **Type of information**: Numeric (float)
- **Units**: °C

Dependent Item Preprocessing
- Preprocessing steps:
  - **Name**: $temp
  - **Parameters**: $temp

© Copyright 2020 IntelliTrend GmbH • Germany • www.intellitrend.de
“Let me subscribe”
Master item and dependent items

<table>
<thead>
<tr>
<th>Wizard</th>
<th>Name</th>
<th>Triggers</th>
<th>Key</th>
<th>Interval</th>
<th>History</th>
<th>Trends</th>
<th>Type</th>
<th>Applications</th>
<th>Status</th>
<th>Info</th>
</tr>
</thead>
<tbody>
<tr>
<td>...</td>
<td>IMS-Smart Sensor MQTT: Sensor data master: Air pressure [Pa]</td>
<td>1</td>
<td>press</td>
<td>1000d</td>
<td>365d</td>
<td>1</td>
<td>90d</td>
<td>Dependent item</td>
<td>Sensor data</td>
<td>Enabled</td>
</tr>
<tr>
<td>...</td>
<td>IMS-Smart Sensor MQTT: Sensor data master: Battery charge [%]</td>
<td>1</td>
<td>batterycharge</td>
<td>1000d</td>
<td>365d</td>
<td>1</td>
<td>90d</td>
<td>Dependent item</td>
<td>Sensor data</td>
<td>Enabled</td>
</tr>
<tr>
<td>...</td>
<td>IMS-Smart Sensor MQTT: Sensor data master: Dew point [C]</td>
<td>2</td>
<td>dew</td>
<td>90d</td>
<td>365d</td>
<td>1</td>
<td>90d</td>
<td>Dependent item</td>
<td>Sensor data</td>
<td>Enabled</td>
</tr>
<tr>
<td>...</td>
<td>IMS-Smart Sensor MQTT: Sensor data master: Equivalent carbon dioxide [CO2] [ppm]</td>
<td>1</td>
<td>co2</td>
<td>90d</td>
<td>365d</td>
<td>1</td>
<td>90d</td>
<td>Dependent item</td>
<td>Sensor data</td>
<td>Enabled</td>
</tr>
<tr>
<td>...</td>
<td>IMS-Smart Sensor MQTT: Sensor data master: Humidity [%]</td>
<td>1</td>
<td>hum</td>
<td>90d</td>
<td>365d</td>
<td>1</td>
<td>90d</td>
<td>Dependent item</td>
<td>Sensor data</td>
<td>Enabled</td>
</tr>
<tr>
<td>...</td>
<td>IMS-Smart Sensor MQTT: Sensor data master: IQ baseline</td>
<td>1</td>
<td>baseline</td>
<td>90d</td>
<td>365d</td>
<td>1</td>
<td>90d</td>
<td>Dependent item</td>
<td>Sensor data</td>
<td>Enabled</td>
</tr>
<tr>
<td>...</td>
<td>IMS-Smart Sensor MQTT: Sensor data master: IQ disabled [Status]</td>
<td>1</td>
<td>iqd</td>
<td>90d</td>
<td>365d</td>
<td>1</td>
<td>90d</td>
<td>Dependent item</td>
<td>Sensor data</td>
<td>Enabled</td>
</tr>
<tr>
<td>...</td>
<td>IMS-Smart Sensor MQTT: Sensor data master: mqqt(\text{payload})[MQTT_BROKER],[MQTT_TOPIC],[MQTT_USER],[MQTT_PASS]</td>
<td>1</td>
<td></td>
<td>90d</td>
<td>365d</td>
<td>1</td>
<td>90d</td>
<td>Zabbix agent (active)</td>
<td>Sensor data</td>
<td>Enabled</td>
</tr>
<tr>
<td>...</td>
<td>IMS-Smart Sensor MQTT: Sensor data master: Temperature external [C]</td>
<td>2</td>
<td>temp</td>
<td>90d</td>
<td>365d</td>
<td>1</td>
<td>90d</td>
<td>Dependent item</td>
<td>Sensor data</td>
<td>Enabled</td>
</tr>
<tr>
<td>...</td>
<td>IMS-Smart Sensor MQTT: Sensor data master: Temperature [C]</td>
<td>1</td>
<td>temp</td>
<td>90d</td>
<td>365d</td>
<td>1</td>
<td>90d</td>
<td>Dependent item</td>
<td>Sensor data</td>
<td>Enabled</td>
</tr>
<tr>
<td>...</td>
<td>IMS-Smart Sensor MQTT: Sensor data master: Total volatile organic compound TVOC [ppb]</td>
<td>1</td>
<td>tvoc</td>
<td>90d</td>
<td>365d</td>
<td>1</td>
<td>90d</td>
<td>Dependent item</td>
<td>Sensor data</td>
<td>Enabled</td>
</tr>
<tr>
<td>...</td>
<td>IMS-Smart Sensor MQTT: Sensor data master: VOC [L]</td>
<td>1</td>
<td>voc</td>
<td>90d</td>
<td>365d</td>
<td>1</td>
<td>90d</td>
<td>Dependent item</td>
<td>Sensor data</td>
<td>Enabled</td>
</tr>
<tr>
<td>...</td>
<td>IMS-Smart Sensor MQTT: Sensor data master: WiFi RSSI [dBm]</td>
<td>1</td>
<td>rssi</td>
<td>90d</td>
<td>365d</td>
<td>1</td>
<td>90d</td>
<td>Dependent item</td>
<td>Sensor data</td>
<td>Enabled</td>
</tr>
</tbody>
</table>

Displaying 13 of 13 found
„Let me subscribe“
Sensor data dashboard
Bonus:
Publish the outcome of a Zabbix trigger, so it can be consumed by other MQTT based devices
„Let me subscribe“
Publish from Zabbix

Simple media type to send problems to the topic:

zabbix/problems/<user>

```
#!/bin/sh
mosquitto_pub -h yourbroker.io -m "$1" -t "zabbix/problems/$2"
```
Let me subscribe
Publish from Zabbix

JSON problem template:

```json
{
  "type": "problem",
  "start-date": "{EVENT.DATE}" ,
  "start-time": "{EVENT.TIME}" ,
  "name": "{EVENT.NAME}" ,
  "host": "{HOST.NAME}" ,
  "severity": "{EVENT.SEVERITY}" ,
  "opdata": "{EVENT.OPDATA}" ,
  "event-id": "{EVENT.ID}" ,
  "trigger-url": "{TRIGGER.URL}" 
}
```

JSON problem recovery template:

```json
{
  "type": "recovery",
  "recovery-date": "{EVENT.RECOVERY.DATE}" ,
  "recovery-time": "{EVENT.RECOVERY.TIME}" ,
  "name": "{EVENT.NAME}" ,
  "host": "{HOST.NAME}" ,
  "duration": "{EVENT.DURATION}" ,
  "severity": "{EVENT.SEVERITY}" ,
  "opdata": "{EVENT.OPDATA}" ,
  "event-id": "{EVENT.ID}" ,
  "trigger-url": "{TRIGGER.URL}" 
}
```
„Let me subscribe“
Publish from Zabbix

Zabbix problems are now published via MQTT.

Any MQTT subscriber, like Node-RED, receives the alert.
„Let me subscribe“
Publish from Zabbix

IoT devices and other subscribers can react to issues detected by Zabbix using Node-RED!
“Let me subscribe”
Try it out – Live broker available

https://github.com/intellitrend/zabbix-iot-mqtt
„Let me subscribe“
Zabbix masters IoT Topics

Thank You!

IntelliTrend IT-Services GmbH

Otto-Brenner-Strasse 119

D-33607 Bielefeld, Germany

Contact: Wolfgang Alper

wolfgang.alper@intellitrend.de

www.intellitrend.de